

07-12 00

H

+

07/07/00
JCE41 U.S. PTO

09644877 070700

NEW UTILITY PATENT APPLICATION TRANSMITTAL <i>(only for new nonprovisional applications under 37 CFR 1.53(b))</i>	Attorney Docket Number	4461
	First Named Inventor	Pawan Goyal
	Total Pages in this Submission	30
	Express Mail Label No.	EL482716793US

09/6/00
JCE41 U.S. PTO

09/07/00

APPLICATION ELEMENTS	ACCOMPANYING APPLICATION PARTS
1. <input checked="" type="checkbox"/> Fee Transmittal Form (in duplicate) <input checked="" type="checkbox"/> Check Enclosed 2. <input checked="" type="checkbox"/> Specification <i>(preferred arrangement set forth below)</i> <input type="checkbox"/> Descriptive Title of the Invention <input type="checkbox"/> Cross Reference(s) to Related Case(s) <input type="checkbox"/> Statement Regarding Fed sponsored R & D <input type="checkbox"/> Background of the Invention <input type="checkbox"/> Brief Summary of the Invention <input type="checkbox"/> Brief Description of the Drawing(s) <input type="checkbox"/> Detailed Description <input type="checkbox"/> Claim or Claims <input type="checkbox"/> Abstract of the Disclosure 3. <input checked="" type="checkbox"/> Drawing(s) <i>(when necessary per 35 USC 113)</i> 4. Oath or Declaration a. <input checked="" type="checkbox"/> New Declaration <input checked="" type="checkbox"/> Executed b. <input type="checkbox"/> Copy from a prior application (37 CFR 1.63(d)) <i>(for continuation/divisional with Box 17 completed)</i> i. <input type="checkbox"/> DELETION OF INVENTOR(S) Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b). 5. <input type="checkbox"/> Incorporation by Reference <i>(useable if Box 4b is checked)</i> . The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.	6. <input type="checkbox"/> Assignment & Assignment Recordation Cover Sheet 7. <input type="checkbox"/> Certified Copy of Priority Document(s) <i>(if foreign priority is claimed)</i> 8. <input type="checkbox"/> Information Disclosure Statement & PTO-1449 <input type="checkbox"/> Copies of IDS Citation(s) 9. <input type="checkbox"/> Preliminary Amendment 10. Small Entity Statement <input checked="" type="checkbox"/> New Statement enclosed <input type="checkbox"/> Statement filed in prior application. Status still proper and desired 11. <input checked="" type="checkbox"/> Return Postcard 12. <input type="checkbox"/> 13. <input type="checkbox"/> 14. <input type="checkbox"/> 15. <input type="checkbox"/> 16. <input type="checkbox"/>

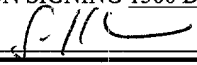
ADDRESS TO:
Box Patent Application
Commissioner for Patents
Washington, D.C. 20231

17. If a **CONTINUING APPLICATION**, check appropriate box and supply the requisite information below and in a preliminary amendment:
☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No: ____/____
 Prior application information: Examiner: _____ Group/Art Unit: _____

18. CORRESPONDENCE ADDRESS					
NAME	Albert C. Smith Fenwick & West LLP				
ADDRESS	Two Palo Alto Square				
CITY	Palo Alto	STATE	CA	ZIP CODE	94306
COUNTRY	U.S.A.	TELEPHONE	(650) 858-7296	FAX	(650) 494-1417
Name (Print/Type)	Albert C. Smith			Registration No. (Attorney/Agent)	20,355
Signature	A. C. Smith			Date	7/7/2000



VERIFIED STATEMENT CLAIMING SMALL ENTITY STATUS (37 CFR 1.9(f) & 1.27(c))--SMALL BUSINESS CONCERN	Docket Number (Optional): 4461
--	--

Applicant or Patentee: <u>Pawan Goyal, Snorri Gylfason, Xun Wilson Huang, Srinivasan Keshav and Rosen Sharma</u>	
Application or Patent No.: <u>not yet known</u>	
Filing Date or Issue Date: <u>not yet known</u>	
Title: <u>ASSOCIATING IDENTIFIERS WITH VIRTUAL PROCESSES</u>	
I hereby declare that I am	
<input type="checkbox"/> the owner of the small business concern identified below:	
<input checked="" type="checkbox"/> an official of the small business concern empowered to act on behalf of the concern identified below:	
NAME OF SMALL BUSINESS CONCERN <u>Ensim Corporation</u>	
ADDRESS OF SMALL BUSINESS CONCERN <u>1366 Borregas Avenue, Sunnyvale, CA 94089</u>	
<p>I hereby declare that the above identified small business concern qualifies as a small business concern as defined in 13 CFR 121.12, and reproduced in 37 CFR 1.9(d), for purposes of paying reduced fees to the United States Patent and Trademark Office, in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average over the previous fiscal year of the concern of the persons employed on a full-time, part-time or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third party or parties controls or has the power to control both.</p> <p>I hereby declare that rights under contract or law have been conveyed to and remain with the small business concern identified above with regard to the invention described in:</p> <p><input checked="" type="checkbox"/> the specification filed herewith with title as listed above.</p> <p><input type="checkbox"/> the application identified above.</p> <p><input type="checkbox"/> the patent identified above.</p> <p>If the rights held by the above identified small business concern are not exclusive, each individual, concern or organization having rights in the invention must file separate verified statements averring to their status as small entities, and no rights to the invention are held by any person, other than the inventor, who would not qualify as an independent inventor under 37CFR 1.9(c) if that person made the invention, or by any concern which would not qualify as a small business concern under 37 CFR 1.9(d), or a nonprofit organization under 37 CFR 1.9(e).</p> <p>Each such person, concern or organization having any rights in the invention is listed below:</p> <p><input checked="" type="checkbox"/> No such person, concern, or organization exists.</p> <p><input type="checkbox"/> Each such person, concern or organization is listed below:</p> <p>_____</p> <p>_____</p> <p>Separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities. (37 CFR 1.27)</p> <p>I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))</p> <p>I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.</p> <p>NAME OF PERSON SIGNING <u>Srinivasan Keshav</u></p> <p>TITLE OF PERSON IF OTHER THAN OWNER <u>Chief Technical Officer</u></p> <p>ADDRESS OF PERSON SIGNING <u>1366 Borregas Avenue, Sunnyvale, CA 94089</u></p> <p>SIGNATURE <u></u> DATE <u>6/29/00</u></p>	

00611877-070700

ASSOCIATING IDENTIFIERS WITH VIRTUAL PROCESSES

5

INVENTORS

Pawan Goyal, Snorri Gylfason, Xun Wilson Huang, Srinivasan Keshav and Rosen
Sharma

10

BACKGROUND

FIELD OF INVENTION

The present invention relates generally to virtual computer processes, and
specifically to associating an identifier with each of a plurality of processes comprising a
15 virtual process.

BACKGROUND OF INVENTION

With the popularity and success of the Internet, server technologies are of great
commercial importance today. Typically, an individual server application executes on a
single physical host computer, servicing client requests. However, providing a unique
20 physical host for each server application is expensive and inefficient. Hosting services
are often provided commercially by an Internet Service Provider (ISP). Typically, an ISP
has to provide a separate physical host computer on which to execute a server application
for every customer that purchases hosting services. Often, a customer purchasing hosting

services from an ISP will neither require nor be amenable to paying for use of an entire host computer. Generally, only a fraction of the processing power, storage, and other resources of a host computer will be required to meet the needs of an individual customer.

5 It is desirable for an ISP to be able to provide multiple, server applications on a single physical host computer. However, in order to be commercially viable, every server application would have to be isolated from every other server application running on the same physical host. Obviously, it would unacceptable to customers of an ISP to purchase hosting services, only to have another server application program (perhaps
10 belonging to a competitor) be able to access the customer's data and client requests. Thus, each server application program would have to be isolated, receiving only requests from its own clients, transmitting data only to its own clients, and being prevented from accessing data associated with other server applications. Furthermore, it would be necessary to allocate varying specific levels of system resources to different server
15 applications, depending upon the needs of and amounts paid by the various customers of the ISP. In effect, each server application would need to comprise a virtual private server, simulating a server application executing on a dedicated physical host computer.

Such functionality has been heretofore impossible because a virtual private server, rather than comprising a single, discrete process, must be made up of a plurality of
20 seemingly unrelated processes, each performing various elements of the sum total of the functionality required by the customer. Because each virtual private server must comprise a plurality of processes, it has been impossible for an ISP to isolate the processes associated with one virtual private server from those processes associated with

other virtual private servers. What is needed is a method whereby the individual ones of pluralities of separate processes comprising multiple, virtual private servers can each be associated with their respective virtual private server.

SUMMARY OF INVENTION

5 The present invention facilitates the association of multiple processes with their respective virtual private servers. In order to run multiple virtual processes on a single, physical computer system, each virtual private server is started by executing a separate, system initialization process. Each system initialization process is associated with a virtual private server identifier, for example by storing the identification number of the
10 process and the virtual private server identifier in a data structure in computer memory. Each virtual private server identifier may be associated with a particular customer to identify the virtual private server belonging to the customer.

 System calls that create processes are intercepted, and a system call wrapper associates created processes with the virtual private server identifier with which the
15 process that made the system call is associated. Thus, all processes originating from each system initialization process will be associated with the virtual private server identifier associated with the corresponding system initialization process. Because all processes that are part of a virtual private server will be created by the associated system initialization process or by its progeny, all processes comprising a virtual private server
20 will be associated with the corresponding virtual private server identifier. This allows all processes that are a part of each customer's virtual private server to be identified and

segregated from the processes of other customers, even though these processes are executing on the same physical server.

It will be readily apparent to one skilled in the art that the present invention can be utilized to associate a plurality of processes comprising any type of virtual process with a corresponding virtual process identifier. Of course, all such utilizations are within the scope of the present invention. Although in one embodiment, the virtual process is in the form of a virtual private server, the present invention is by no means limited to this embodiment.

The features and advantages described in this summary and the following detailed description are not all-inclusive, and particularly, many additional features and advantages will be apparent to one of ordinary skill in the art in view of the drawings, specification, and claims hereof. Moreover, it should be noted that the language used in the specification has been principally selected for readability and instructional purposes, and may not have been selected to delineate or circumscribe the inventive subject matter, resort to the claims being necessary to determine such inventive subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram providing a high level overview of a system for associating identifiers with virtual processes, according to one embodiment of the present invention.

FIG. 2 is a block diagram illustrating a system utilizing a virtual process manager program, according to one embodiment of the present invention.

FIG. 3 is a block diagram illustrating a system utilizing a modified loader program, according to another embodiment of the present invention.

The figures depict embodiments of the present invention for purposes of illustration only. One skilled in the art will readily recognize from the following discussion that alternative embodiments of the structures and methods illustrated herein may be employed without departing from the principles of the invention described herein.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

I. SYSTEM OVERVIEW

FIG. 1 illustrates a high level overview of a system 100 for associating identifiers with virtual processes according to one embodiment of the present invention. A computer memory 101 includes user address space 103 and operating system address space 105. Multiple initialization processes 107 execute in user address space 103. Although FIG. 1 illustrates only two initialization processes 107 executing in user address space 103, it is to be understood that within a given computer memory 101, more than two initialization processes 107 can execute simultaneously.

Also executing in user address space are descendent processes 108, originating from the initialization processes 107. A descendent process 108 is a child process of an initialization process 107, or a child process thereof, extended to any number of generations of subsequent child processes. Although FIG. 1 illustrates only two descendent processes 108 for each initialization process 107, it is to be understood that within a given computer memory 101, fewer or more than two descendent processes 108 per initialization process 107 can execute simultaneously.

Preferably, a data structure for storing associations 129 between executing processes (initialization processes 107 or descendent processes 108) and virtual processes

is inserted into the operating system 117. In one embodiment, the data structure is a mapping table 127, but in other embodiments other data structures are utilized, for example a linked list. In one embodiment, the mapping table 127 (or other data structure) is dynamically loaded into the operating system kernel 109, while the kernel 109 is active. In another embodiment, the mapping table 127 is stored in user address space 103. The maintenance and use of the mapping table 127 is discussed in detail below.

It is to be understood that a virtual process is not an actual process that executes in computer memory 101. Instead, the term “virtual process” describes a collection of associated functionality. For example, a virtual private server application is not actually a discrete process, but instead comprises a plurality of actual processes that together provide the desired functionality, thereby simulating the existence of a single server application executing on a dedicated physical host. Each actual process that performs some of the functionality of the virtual private server is a part of the virtual process. In FIG. 1 for example, initialization process 1 and descendent processes 1 and 2 (the processes descended from initialization process 1) comprise one virtual process, whereas initialization process 2 and descendent processes 3 and 4 comprise another.

In order to associate a specific identifier with each actual process that is a part of a virtual process, a separate system initialization process 107 is started for each virtual process. Normally, each process executing on a multitasking operating system is descended from a single system initialization process 107 that is started when the operating system 117 is booted. However, the present invention uses techniques described in detail below to start a separate system initialization process 107 for each virtual process. When each system initialization process 107 is started, an association

129 between the system initialization process 107 and the virtual process is stored in the mapping table 127. All additional processes that are descended from a given initialization process are thus identified with the virtual process associated with that initialization process.

5 In one embodiment, rather than starting a separate system initialization process 107 for each virtual process, a custom initialization process is started. In this embodiment, all processes that are a part of a specific virtual process are descended from the associated custom initialization process, and are associated with the virtual process with which the custom initialization process is associated. The exact functionality
10 included in the custom initialization process is a design choice that can be made by a system administrator.

System calls 115 that generate child processes (for example, the UNIX® fork and clone functions) are intercepted so that the child processes can be associated with the virtual process with which the parent process is associated. A system call wrapper 111 is
15 utilized in order to intercept system calls 115. In one embodiment, the system call wrapper 111 is dynamically loaded into the operating system kernel 109, while the kernel 109 is active. In another embodiment, the system call wrapper is loaded in user address space 103. The system call wrapper 111 is preferably in the form of object code, the functional features of which are described in detail below.

20 Pointers 114 to system calls 115 are located in an operating system call vector table 113. It is to be understood that the term "system call vector table" as used herein denotes an area in operating system address space 105 in which there are stored the addresses of system calls. In the UNIX® operating system, this part of the operating

system is called the "system call vector table," and that term is used in this specification.

Other operating systems employ different terminology to denote the same system component. A system call vector table by any other name is still within the scope of the present invention.

5 A copy 116 is made of a pointer 114 to each system call 115 to be intercepted. These copies 116 of pointers 114 are preferably stored in operating system address space 105, but in an alternative embodiment are stored in user address space 103. Once the copies 116 have been made and saved, the pointers 114 in the system call vector table 113 to the system calls 115 to be intercepted are replaced with pointers 118 to the system
10 call wrapper 111, such that when a system call 115 to be intercepted is made, the system call wrapper 111 executes instead. In one embodiment, this copying, storing, and replacing of pointers is performed by the system call wrapper 111. In other embodiments, copying, storing, and replacing of pointers is performed by a pointer management module executing in either operating system address space 105 or user
15 address space 103 as desired. The pointer management module can either be a stand alone program, or a component of a larger application program as desired.

 Executing alternative code when a system call 115 is made comprises intercepting the system call 115. The steps of inserting a system call wrapper 111 into the operating system 117, making a copy 116 of an operating system pointer 114 to a system call 115,
20 and replacing the operating system pointer 114 with a pointer 118 to the system call wrapper 111 facilitate interception of a system call 115. When a system call 115 to be intercepted is made, the operating system 117 uses the pointer 118 in the system call vector table 113 to the system call wrapper 111 to execute the system call wrapper 111.

It is to be understood that only system calls 115 that create child processes need be intercepted, and thus only pointers 114 to system calls 115 to be intercepted are replaced with pointers 118 to the system call wrapper 111. Pointers 114 to system calls 115 which are not to be intercepted are not replaced. Thus, when a non-intercepted
5 system call 115 is made, the system call 115 executes, not the system call wrapper 111.

The various initialization processes 107 and descendent processes 108 execute in user address space 103 under control of the operating system 117, and make system calls 115. When a process makes a system call 115 that creates a child process, the system call wrapper 111 reads the mapping table 127, and determines whether the process that made
10 the system call (the parent of the child process being created) is associated with a virtual process. If so, the system call wrapper 111 uses the saved copy of the pointer 116 to execute the system call 115, allowing the creation of the child process. The system call wrapper 111 updates the mapping table 127, storing an association 129 between the newly created child process and the virtual process with which the process that made the
15 system call is associated. Thus, all descendent processes 108 are associated with the virtual process with which their parent process is associated.

II. STARTING INITIALZIATION PROCESSES BY A MANAGER PROGRAM

FIG. 2 illustrates one embodiment of a system 200 for associating identifiers with
20 virtual processes. In the embodiment illustrated by FIG. 2, the initialization processes 107 are started by a virtual process manager program 201 executing in user address space 103.

The virtual process manager program 201 modifies the operating system 117 of the computer to include the mapping table 127. Preferably, the manager program 201 loads the mapping table 127 into the kernel 109 of the operating system 117 while the kernel is active.

5 For each virtual process, the manager program 201 starts an initialization process 107 from which all other processes that are part of the virtual process will originate as descendent processes 108. Each time the manager program 201 starts an initialization process 107 for a virtual process, the manager program 201 stores, in the mapping table 127, an association 129 between the initialization process 107 and the appropriate virtual
10 process. Subsequently, all additional processes that are part of the virtual process will be originated from the initialization process, and thus associated with the appropriate virtual process.

For example, suppose that the manager program 201 starts a first virtual private server. To do so, the manager process 107 starts an initialization process 107 for the
15 virtual private server, storing an association 129 between the initialization process 107, and a virtual process identifier for the virtual private server. Additional processes that are part of the virtual private server originate from the initialization process 107, and are associated with the virtual process identifier of the virtual private server. The manager process 201 can proceed to start a second virtual private server by starting a separate
20 initialization process 107, and associating the second initialization process 107 with a separate virtual process identifier for the second virtual private server. Consequently, all of the processes associated with the second virtual private server will be associated with

the appropriate virtual process identifier. In this manner, multiple virtual processes on the same physical computer are each associated with unique identifiers.

III. STARTING INITIALZIATION PROCESSES BY A LOADER PROGRAM

5 FIG. 3 illustrates another embodiment of a system 300 for associating identifiers with virtual processes. In the embodiment illustrated by FIG. 3, initialization processes 107 are not started by a manager program 201, but instead are loaded by a modified loader program 301.

10 A loader program is an operating system utility that is used to execute computer programs that are stored on static media. Typically, a loader program loads an executable image from static media into user address space 103 of computer memory 101, and then initiates execution of the loaded image by transferring execution to the first instruction thereof.

15 Like a standard loader program, the modified loader 301 loads executable images (in this case, initialization processes 107) from static media into user address space 103. Additionally, the modified loader program 301 stores, in the mapping table 127, an association 129 between the initialization process 107 being loaded and the appropriate virtual process. Thus, for each virtual process, an initialization process 107 is loaded by the modified loader program, and an association between the initialization process 107
20 and the virtual process is stored in the mapping table 127. Subsequently, additional processes that are part of the virtual process originate from the associated initialization process 107, and are thus associated with the virtual process as described above.

In another embodiment, the modified loader program 301 loads all processes that are part of each virtual process. In that embodiment, whenever the modified loader program 301 loads a process, the modified loader program 301 also stores, in the mapping table, an association 129 between the loaded process and the appropriate virtual process.

As will be understood by those familiar with the art, the invention may be embodied in other specific forms without departing from the spirit or essential characteristics thereof. Likewise, the particular naming of the modules, features, attributes or any other aspect is not mandatory or significant, and the mechanisms that implement the invention or its features may have different names or formats.

Accordingly, the disclosure of the present invention is intended to be illustrative, but not limiting, of the scope of the invention, which is set forth in the following claims.

1 What is claimed is:

- 1 1. A method in a computer system for associating identifiers with virtual
2 processes, the method comprising:
3 for each virtual process, starting a separate first process;
4 associating each first process with a separate virtual process identifier;
5 for each virtual process, originating additional processes included in the
6 virtual process from the first process; and
7 associating all processes that originate from each first process with the
8 associated virtual process identifier.
- 1 2. The method of claim 1 wherein:
2 each virtual process comprises a virtual private server, the virtual private
3 server comprising a plurality of processes that together provide the
4 functionality of a dedicated server application program.
- 1 3. The method of claim 1 wherein:
2 the first process comprises a system initialization process.
- 1 4. The method of claim 1 wherein associating a first process further comprises:
2 storing an entry in a data structure in computer memory, the entry
3 comprising a virtual process identifier and a process identification
4 number of the first process.
- 1 5. The method of claim 1 further comprising:
2 intercepting system calls that create processes; and

5 storing, by the modified loader program, an entry in a data structure in
6 computer memory, the entry comprising a virtual process identifier
7 and a process identification number of the first process.

1 10. The method of claim 1 further comprising:

2 starting the first process by a virtual process manager program; and
3 storing, by the manager program, an entry in a data structure in computer
4 memory, the entry comprising a virtual process identifier and a
5 process identification number of the first process.

1 11. A computer program product for associating identifiers with virtual
2 processes, the computer program product comprising:

3 program code for starting, for each virtual process, a separate first process;
4 program code for associating each first process with a separate virtual
5 process identifier;
6 program code for originating, from each virtual process, additional
7 processes included in the virtual process;
8 program code for associating all processes that originate from each first
9 process with the associated virtual process identifier; and
10 a computer readable medium on which the program codes are stored.

1 12. The computer program product of claim 11 further comprising:

2 program code for storing an entry in a data structure in computer memory,
3 the entry comprising a virtual process identifier and a process
4 identification number of the first process.

1 13. The computer program product of claim 11 further comprising:

2 program code for intercepting system calls that create processes; and

3 program code for associating a process being created with the virtual

4 process identifier of a process that made the system call.

1 14. The computer program product of claim 11 further comprising:

2 program code comprising instructions to associate the process being

3 created with the virtual process identifier of the process that made

4 the system call; and

5 program code for replacing a pointer to the system call with a pointer to

6 the instructions to associate, such that calling the system call

7 causes the instructions to associate to execute.

1 15. The computer program product of claim 14 further comprising:

2 program code for storing the instructions to associate in the operating

3 system by loading a module into a running operating system

4 kernel, the module comprising the instructions to associate.

1 16. The computer program product of claim 11 further comprising:

2 program code for starting the first process; and

3 program code for storing an entry in a data structure in computer memory,

4 the entry comprising a virtual process identifier and a process

5 identification number of the first process.

1 17. The computer program product of claim 16 further comprising:

2 program code for loading the first process into computer memory.

1 18. A computer system for associating identifiers with virtual processes, the
2 computer system comprising:

3 an initiation module, for starting a separate first process for each virtual
4 process;

5 an origination module, coupled to the initiation module, for originating,
6 from each virtual process, additional processes included in the
7 virtual process; and

8 an association module, coupled to the origination module, for associating
9 each first process with a separate virtual process identifier, and for
10 associating all processes that originate from each first process with
11 the associated virtual process identifier.

1 19. The system of claim 18 further comprising:

2 a storage module, coupled to the association module, for storing an entry
3 in a data structure in computer memory, the entry comprising a
4 virtual process identifier and a process identification number of the
5 first process.

1 20. The system of claim 18 further comprising:

2 an interception module, coupled to the association module, for
3 intercepting system calls that create processes; and wherein

4 the association module is further for associating a process being created
5 with the virtual process identifier of a process that made the system
6 call.

1 21. The system of claim 20 further comprising:

2 a storage module, coupled to the interception module, for storing object
3 code comprising instructions to associate the process being created
4 with the virtual process identifier of the process that made the
5 system call; and wherein
6 the interception module is further for replacing a pointer to the system call
7 with a pointer to the stored object code, such that calling the
8 system call causes the object code to execute.

1 22. The system of claim 21 further comprising:

2 coupled to the storage module, an insertion module, for inserting the
3 instructions to associate in the operating system by loading a
4 module into a running operating system kernel.

1 23. A method in a computer system for associating identifiers with virtual
2 processes, the method comprising:
3 starting each virtual process by executing a separate, system initialization
4 process;
5 associating each system initialization process with a virtual process
6 identifier;

7 associating all processes originating from each system initialization
8 process with the virtual process identifier associated with the
9 system initialization process; and wherein
10 a virtual process is comprised of a single system initialization process and
11 all processes originating therefrom.

1 24. A method in a computer system for associating identifiers with virtual
2 processes:

3 associating initialization processes with respective virtual process
4 identifiers; and
5 in response to a process being created from one of the initialization
6 processes, associating the process created by the initialization
7 process with the virtual process identifier associated with the
8 initialization process.

1 25. A method in a computer system for associating identifiers with virtual
2 processes, the method comprising:

3 loading each process that is a part of a virtual process into computer
4 memory by a modified loader program; and
5 storing, by the modified loader program, an entry in a data structure in
6 computer memory, the entry comprising a virtual process identifier
7 and a process identification number of the loaded process.

ASSOCIATING IDENTIFIERS WITH VIRTUAL PROCESSES

ABSTRACT OF THE DISCLOSURE

Multiple virtual processes are each associated with a unique virtual process

5 identifier. For each virtual process, a separate initialization process is started, from which
all other processes included in the virtual process originate. The initialization process is
associated with the appropriate virtual process identifier. System calls that create new
processes are intercepted, and created processes are associated with the virtual process
identifier of the appropriate initialization process. Multiple virtual processes are provided
10 on a single, physical computer system.

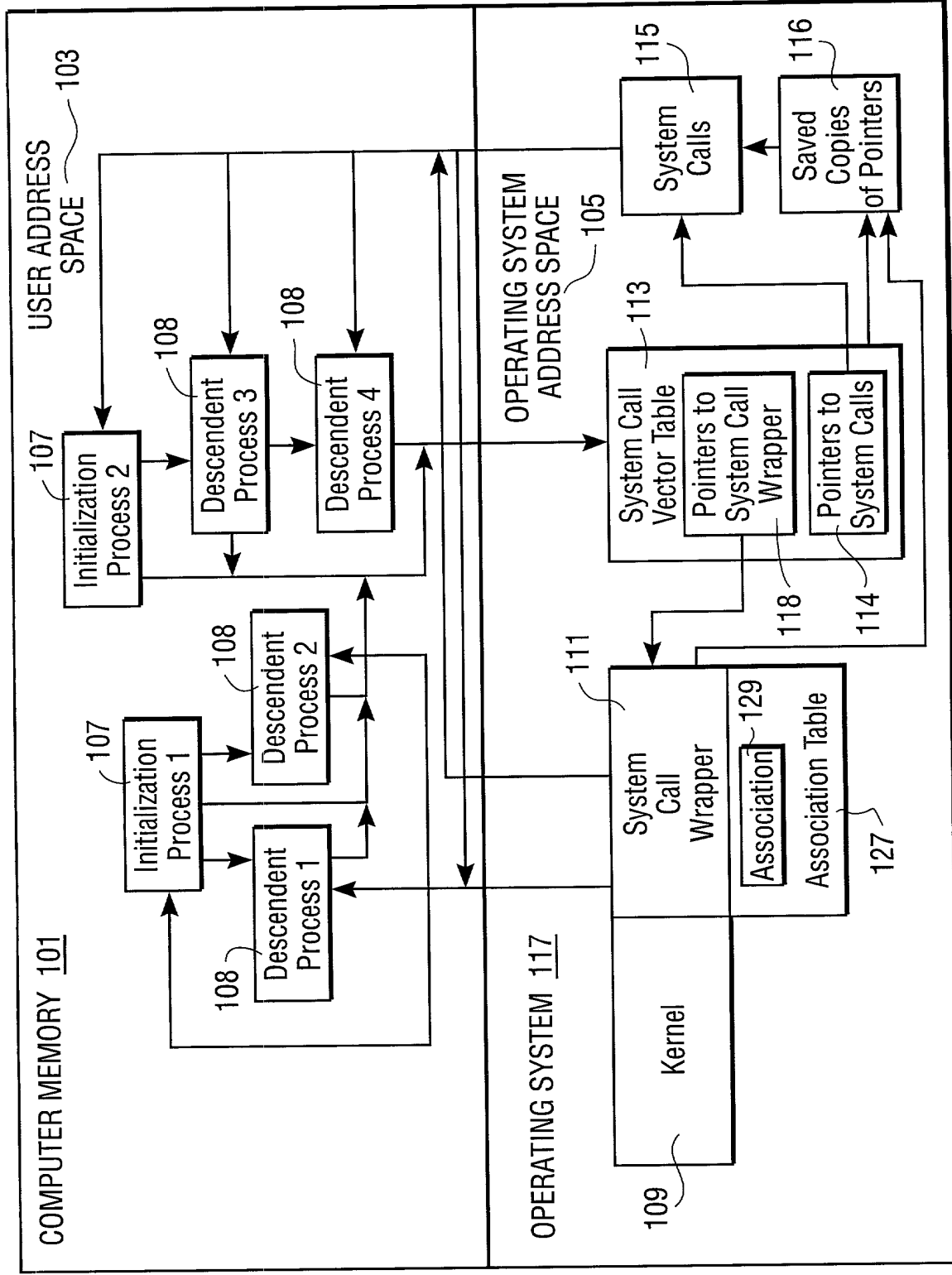


FIG. 1

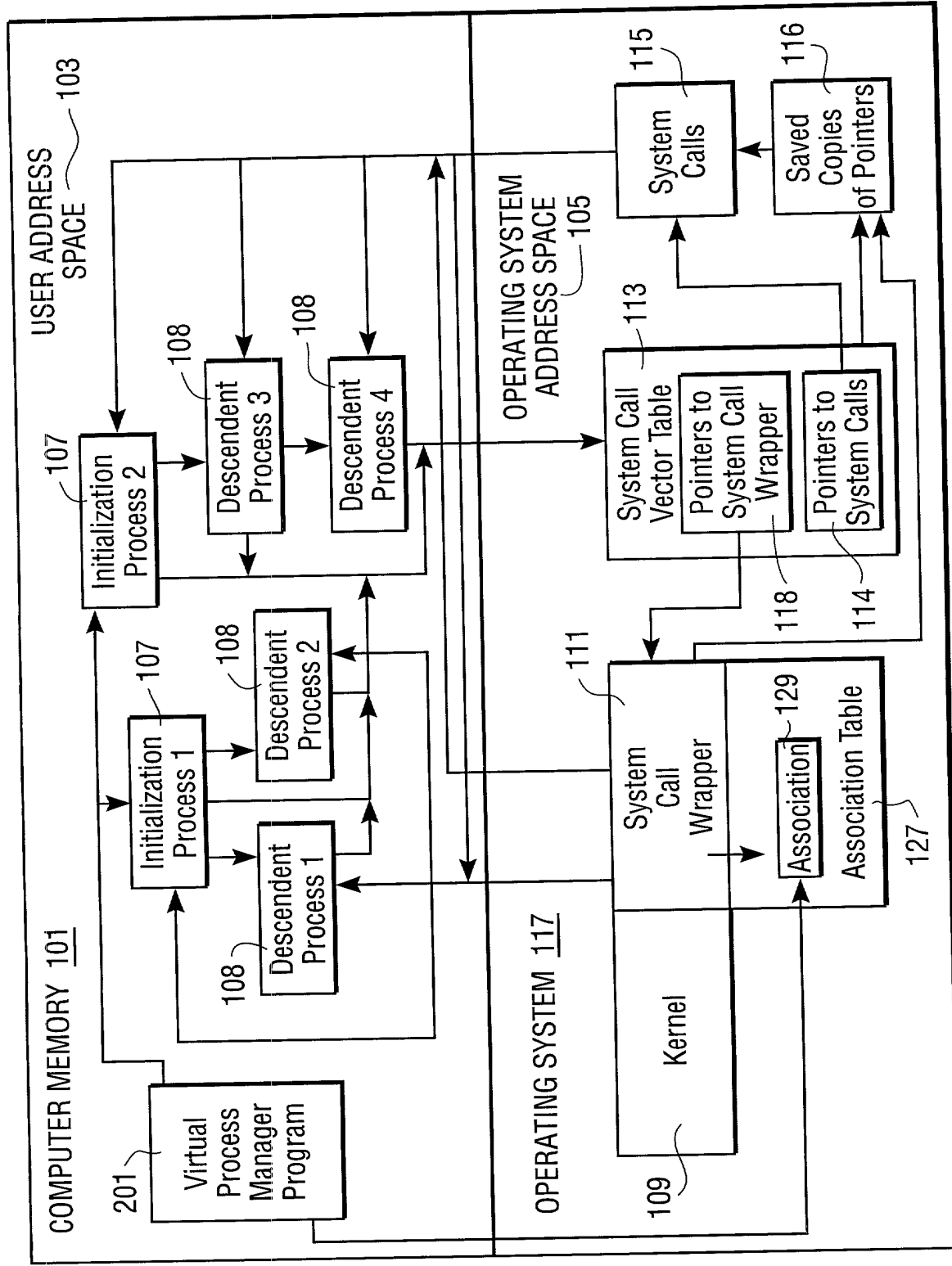


FIG. 2

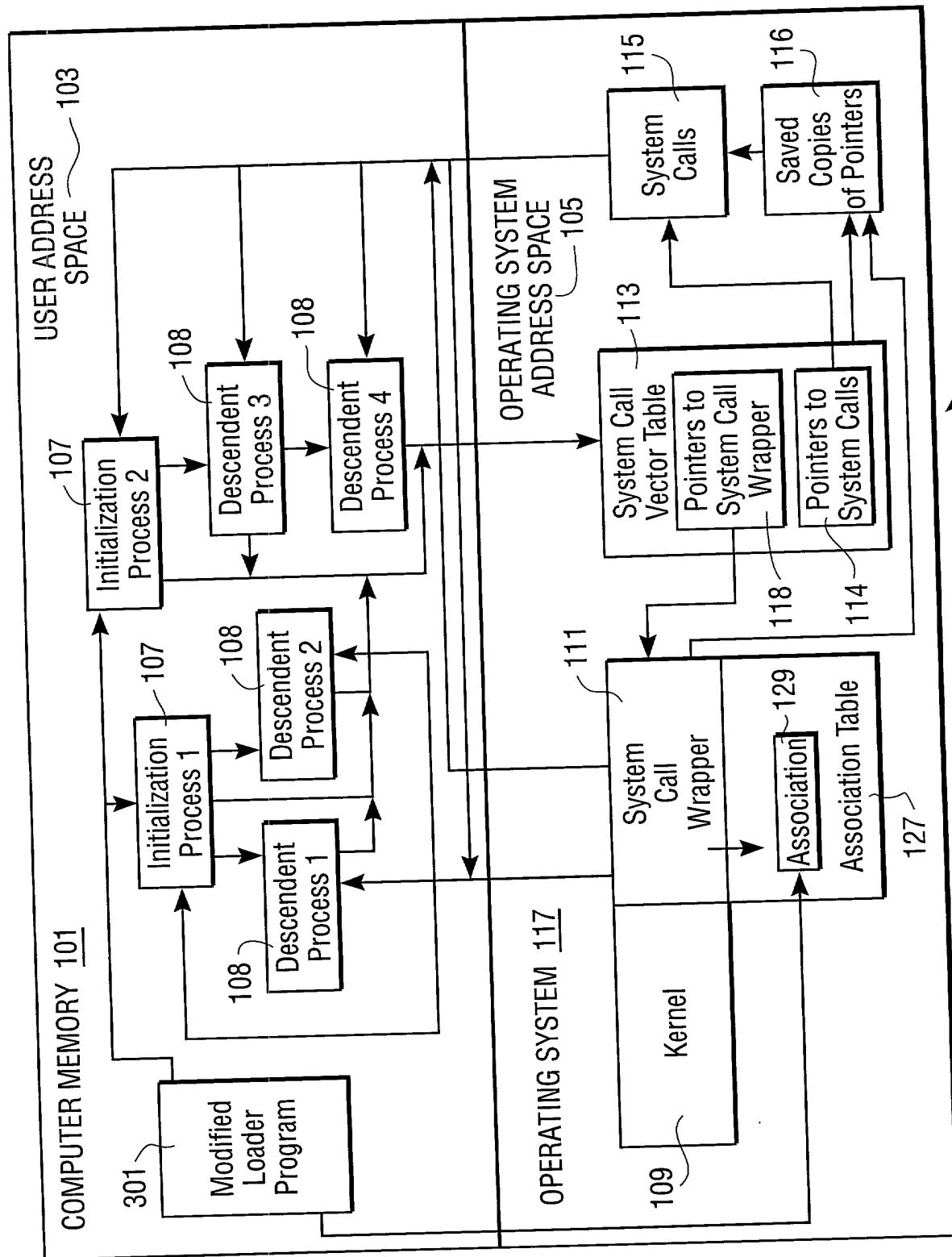


FIG. 3

0010/PTO Rev. 6/95 U.S. Department of Commerce Patent and Trademark Office DECLARATION FOR UTILITY OR DESIGN PATENT APPLICATION <input checked="" type="checkbox"/> Declaration Submitted with Initial Filing OR <input type="checkbox"/> Declaration Submitted after Initial Filing	Attorney Docket Number	4461
	First Named Inventor	Pawan Goyal
	<i>COMPLETE IF KNOWN</i>	
	Application Number	not yet known
	Filing Date	not yet known
	Group Art Unit	not yet known
Examiner Name	not yet known	

As a below named inventor, I hereby declare that:

My residence, post office address, and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

ASSOCIATING IDENTIFIERS WITH VIRTUAL PROCESSES

the specification of which *(Title of the Invention)*

☒ is attached hereto
OR

☐ was filed on (MM/DD/YYYY) [] as United States Application Number or PCT International Application Number [] and was amended on (MM/DD/YYYY) [] (if applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment specifically referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37 Code of Federal Regulations. § 1.56.

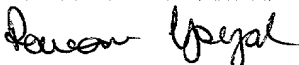
I hereby claim foreign priority benefits under Title 35, United States Code § 119 (a)-(d) or § 385(b) of any foreign application(s) for patent or inventor's certificate, or § 365 (a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or of any PCT international application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application Number(s)	Country	Foreign Filing Date (MM/DD/YYYY)	Priority Not Claimed	Certified Copy Attached?	
				YES	NO
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

☐ Additional foreign application numbers are listed on a supplemental priority sheet attached hereto:


I hereby claim the benefit under Title 35, United States Code § 119(e) of any United States provisional application(s) listed below.

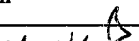
Application Number(s)	Filing Date (MM/DD/YYYY)	<input type="checkbox"/> Additional provisional application numbers are listed on a supplemental sheet attached hereto.


DECLARATION				Page 2	
<p>I hereby claim the benefit under Title 35, United States Code § 120 of any United States application(s), or § 365(c) of any PCT international application designating the United States of America, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT international application in the manner provided by the first paragraph of Title 35, United States Code § 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations § 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application.</p>					
U.S. Parent Application Number	PCT Parent Number	Parent Filing Date (MM/DD/YYYY)	Parent Patent Number (if applicable)		
<input type="checkbox"/> Additional U.S. or PCT international application numbers are listed on a supplemental priority sheet attached hereto.					
<p>As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith:</p>					
Name		Registration Number	Name		Registration Number
Albert C. Smith Laura A. Majerus		20,355	Robert R. Sachs Renée M. DuBord		42,120
		33,417			42,500
<input type="checkbox"/> Additional attorney(s) and/or agent(s) named on a supplemental sheet attached hereto.					
<p>Please direct all correspondence to:</p> <p style="text-align: center;"> Robert R. Sachs Fenwick & West LLP Two Palo Alto Square Palo Alto, CA 94306 U.S.A. </p>					
Telephone	(650) 858-7110		Fax	(650) 494-1417	
<p>I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.</p>					
Name of Sole or First Inventor:		<input type="checkbox"/> A petition has been filed for this unsigned inventor			
Given Name	Pawan	Middle Initial		Family Name	Goyal
Inventor's Signature				Date	6/30/00
Residence: City	Mountain View	State	CA	Country	U.S.A.
Citizenship	India				
Mailing Address	777 W. Middlefield Road, #83				
Mailing Address					
City	Mountain View	State	CA	Zip	94043
Country	U.S.A.				
<input checked="" type="checkbox"/> Additional inventors are being named on supplemental sheet(s) attached hereto					


09611877-070700

+

DECLARATION					ADDITIONAL INVENTOR(S) Supplemental Sheet					
Name of Additional Joint Inventor, if any:				[] A petition has been filed for this unsigned inventor						
Given Name	Snorri			Middle Initial		Family Name	Gylfason		Suffix e.g. Jr.	
Inventor's Signature							Date	6/30/00		
Residence: City	Mountain View			State	CA	Country	U.S.A.		Citizenship	Iceland
Mailing Address	1950 Montecito Avenue, Apt. 20									
Mailing Address										
City	Mountain View			State	CA	Zip	94043		Country	U.S.A.

Name of Additional Joint Inventor, if any:				<input type="checkbox"/> A petition has been filed for this unsigned inventor			
Given Name	Xun Wilson		Middle Initial		Family Name	Huang	Suffix e.g. Jr.
Inventor's Signature						Date	7/1/00
Residence: City	Mountain View		State	CA	Country	U.S.A.	Citizenship PRC
Mailing Address	1950 Montecito Avenue, Apt. 20						
Mailing Address							
City	Mountain View		State	CA	Zip	94043	Country U.S.A.

Name of Additional Joint Inventor, if any:				<input type="checkbox"/> A petition has been filed for this unsigned inventor			
Given Name	Srinivasan		Middle Initial		Family Name	Keshav	Suffix e.g. Jr.
Inventor's Signature						Date	6/29/00
Residence: City	Mountain View		State	CA	Country	U.S.A.	Citizenship India
Mailing Address	834 Sutter Avenue						
Mailing Address							
City	Mountain Veiw		State	CA	Zip	94303	Country U.S.A.

Name of Additional Joint Inventor, if any:										<input type="checkbox"/> A petition has been filed for this unsigned inventor									
Given Name		Rosen				Middle Initial				Family Name		Sharma				Suffix e.g. Jr.			
Inventor's Signature										Date		6/30/00							
Residence: City		Mountain View				State		CA		Country		U.S.A.				Citizenship		India	
Mailing Address		750 N. Shoreline Boulevard, #112																	
Mailing Address																			
City		Mountain View				State		CA		Zip		94043				Country		U.S.A.	
<input type="checkbox"/> Additional inventors are being named on supplemental sheet(s) attached hereto																			